PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-137605
(43)Date of publication of application : 16.05.2000

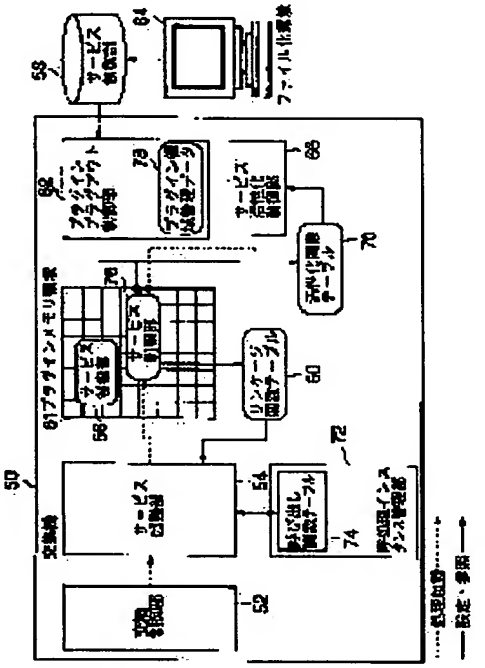
(51)Int.Cl. G06F 9/06
H04M 3/42
H04Q 3/545

(21)Application number : 10-312181
(22)Date of filing : 02.11.1998

(71)Applicant : FUJITSU LTD
(72)Inventor : ISHITA FUMIYASU
YAMADA HIROTOSHI
KAMURO SHINGO

(54) EXCHANGE SERVICE CONTROL METHOD, EXCHANGE AND RECORDING MEDIUM RECORDING EXCHANGE SERVICE CONTROL PROGRAM

(57)Abstract:
PROBLEM TO BE SOLVED: To add/change/delete exchange service without stopping an exchange during an operation by referring to a pointer held by a second table, executing a function which the pointer shows and supplying service.
SOLUTION: A service control part is loaded and the entry address of activation program is recorded in an activation table 70. A service activation control part 66 refers to the entry address recorded in an activation function table 70 and starts a service control part 76. The service control part 76 records the pointer of a function executing service in a linkage function table 60. The pointer of the function obtained by referring to the linkage function table 60 is recorded in the call function table 74 of a call processing instance management part 72 by a service starting part 54. The service starting part 54 refers to the function pointer recorded in the call function table 74, executes the function in the service control part and executes service.



LEGAL STATUS

[Date of request for examination]	22.12.2000
[Date of sending the examiner's decision of rejection]	01.11.2005
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2000－137605
（P2000－137605A）
(43)公開日　平成12年５月16日(2000.5.16)

(51)Int.Cl.'	識別記号	ＦＩ		テーマコード'（参考）
Ｇ０６Ｆ　9/06	５４０	Ｇ０６Ｆ　9/06	５４０Ｆ	５Ｂ０７６
Ｈ０４Ｍ　3/42		Ｈ０４Ｍ　3/42	Ｚ	５Ｋ０２４
Ｈ０４Ｑ　3/545		Ｈ０４Ｑ　3/545		５Ｋ０２６
				９Ａ００１

審査請求　未請求　請求項の数17　ＯＬ　（全 20 頁）

(21)出願番号　特願平10－312181

(22)出願日　平成10年11月２日(1998.11.2)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　位下　史恭
神奈川県横浜市港北区新横浜３丁目９番18号　富士通コミュニケーション・システムズ株式会社内

(72)発明者　山田　博敏
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

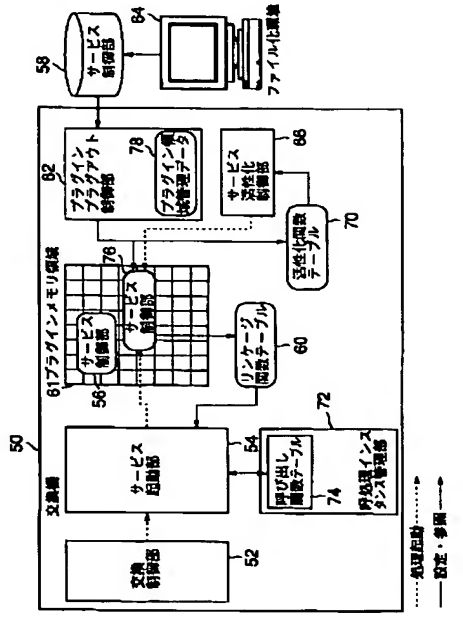
(74)代理人　100070150
弁理士　伊東　忠彦

最終頁に続く

(54)【発明の名称】　交換機サービス制御方法、交換機、及び交換機サービス制御プログラムを記録した記録媒体

(57)【要約】
【課題】稼働中の交換機を停止させることなく、更にサービス実施中における呼び出し中・通話中等の呼状態に影響を与えずに、交換サービスを追加・変更・削除する交換機サービス制御方法、交換機及び記録媒体を提供することを目的とする。
【解決手段】ローディングされたサービスプログラムが、該サービスプログラムが有する関数へのポインタを第１のテーブルに記録し、サービスの利用要求があると呼毎に一時的な記憶領域を確保し、該第１のテーブルを参照して得た該ポインタを該記憶領域内の第２のテーブルに保持し、該第２のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供するよう構成する。

本発明の実施例における交換機の機能構成概要を示す機能ブロック図



1

【特許請求の範囲】
【請求項1 】プログラムにより制御される交換機において、
特定のサービスを実行するサービスプログラムがローディングされると、
所定の場合に該サービスプログラムは該サービスプログラムが有する該サービスを実行するための関数へのポインタを第１のテーブルに記録し、
該交換機は、
発呼により該サービスの利用要求があると呼毎に一時的な記憶領域を確保し、
該第１のテーブルを参照して得た該ポインタを該記憶領域内の第２のテーブルに保持し、
該第２のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供することを特徴とする交換機サービス制御方法。
【請求項２】前記交換機は、
前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを有し、
該データを用いて前記ローディングを制御する請求項１記載の交換機サービス制御方法。
【請求項３】前記所定の場合は、
前記交換機が周期的に前記ローディングを監視して前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である請求項１記載の交換機サービス制御方法。
【請求項４】前記サービスプログラムは独立してコンパイルされて前記ローディングされる請求項１記載の交換機サービス制御方法。
【請求項５】既存のサービスプログラムを別のサービスプログラムに置き換える際に該既存のサービスプログラムを実行中の第１の呼が有る場合、
該別のサービスプログラムによるサービスを提供すべき第２の呼が発生すると、
前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持し、
該ポインタの示す関数を実行することにより該第２の呼に該別のサービスプログラムによるサービスを提供する請求項１記載の交換機サービス制御方法。
【請求項６】ローディングされた前記サービスプログラムを前記交換機から削除する際、
前記第１のテーブルの値を該サービスプログラムがローディングされる前の値に戻す請求項１記載の交換機サービス制御方法。
【請求項７】プログラムにより制御される交換機であって、
特定のサービスを実行するサービスプログラムをローディングする手段と、

2

該サービスプログラムが有する該サービスを実行するための関数へのポインタを保持する第１のテーブルを有し、
所定の場合に該サービスプログラムに該関数へのポインタを該第１のテーブルに記録させる手段と、
発呼により該サービスの利用要求がある場合に呼毎に一時的な記憶領域を確保する手段と、
該第１のテーブルを参照して得た該ポインタを該記憶領域内の第２のテーブルに保持する手段と、
該第２のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供する手段とを有することを特徴とする交換機。
【請求項８】前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを有し、
該データを用いて前記ローディングを制御する手段を有する請求項７記載の交換機。
【請求項９】前記交換機は周期的に前記ローディングを監視する手段とコマンド入力手段を有し、
前記所定の場合は、
該監視する手段により前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である請求項７記載の交換機。
【請求項１０】既存のサービスプログラムを別のサービスプログラムに置き換える際に既存のサービスプログラムを実行中の第１の呼が有る場合、
該別のサービスプログラムによるサービスを提供すべき第２の呼が発生した場合に前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持する手段と、
該ポインタの示す関数を実行することにより該第２の呼に該別のサービスプログラムによるサービスを提供する手段とを有する請求項７記載の交換機。
【請求項１１】ローディングされた前記サービスプログラムを前記交換機から削除する場合に前記第１のテーブルの値を該サービスプログラムがローディングされる前の値に戻す手段を有する請求項７記載の交換機。
【請求項１２】プログラムにより制御される交換機のサービスを制御するコンピュータ読み取り可能な交換機サービス制御プログラムを記録した記録媒体であって、
該交換機に特定のサービスを実行するサービスプログラムをローディングし、
所定の場合に該サービスプログラムに該サービスプログラムが有する該サービスを実行するための関数へのポインタを第１のテーブルに記録させ、
発呼により該サービスの利用要求があると呼毎に一時的な記憶領域を確保し、
該第１のテーブルを参照して得た該ポインタを該記憶領

域内の第２のテーブルに保持し、
該第２のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供することを特徴とする交換機サービス制御プログラムを記録した記録媒体。
【請求項１３】前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを保持し、
該データを用いて前記ローディングを制御する請求項１２記載の交換機サービス制御プログラムを記録した記録媒体。
【請求項１４】前記所定の場合は、
前記交換機が周期的に前記ローディングを監視して前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である請求項１２記載の交換機サービス制御プログラムを記録した記録媒体。
【請求項１５】既存のサービスプログラムを別のサービスプログラムに置き換える際に既存のサービスプログラムを実行中の第１の呼が有る場合、
該別のサービスプログラムによるサービスを提供すべき第２の呼が発生すると、
前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持し、
該ポインタの示す関数を実行することにより該第２の呼に該別のサービスプログラムによるサービスを提供する請求項１２記載の交換機サービス制御プログラムを記録した記録媒体。
【請求項１６】ローディングされた前記サービスプログラムを前記交換機から削除する際、
前記第１のテーブルの値を該サービスプログラムがローディングされる前の値に戻す請求項１２記載の交換機サービス制御プログラムを記録した記録媒体。
【請求項1 ７】プログラムにより制御される交換機のサービスを制御するコンピュータ読み取り可能な交換機サービス制御プログラムを記録した記録媒体であって、
該交換機に特定のサービスを実行させる関数と、
該交換機にローディングされた場合に該関数へのポインタを該交換機内のテーブルに記録するプログラムを含み、
該交換機により該テーブルに記録された該ポインタが参照されて該関数が実行されることを特徴とする交換機サービス制御プログラムを記録した記録媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、交換機にサービス機能を追加等する技術に係り、特に、交換機が提供中のサービスを中断させることなくサービス機能を追加・変更・削除する交換機サービス制御方法及び交換機及び交換機サービス制御プログラムを記録した記録媒体に関する。

【０００２】近年、通信市場開放に伴い通信事業者間の競争が激化しており、通信事業者は加入者への提供サービスにより差別化を図っている。しかし、サービスが多様化するにつれ、そのサービスを実現する交換機におけるソフトウェアの複雑さが一層増したため、短期間でかつ容易に新しいサービスを提供できない現状がある。したがって、特に通信事業者において、新規サービスを導入する場合であっても運転を停止することはもとより、既存サービスへの影響を与えずに新規サービスの機能を追加できる交換機が求められている。
【０００３】
【従来の技術】図１は、従来技術による交換機の交換ソフトウェア実行モジュール１０を示すモジュール図である。同図に示すように、交換ソフトウェア実行モジュール１０はプログラム基盤となるオペレーティングシステム２０、共通機能プログラムモジュール群１６、保守運用プログラムモジュール群１８、交換制御プログラムモジュール群１４、サービス制御プログラムモジュール群１２から構成される。共通機能プログラムモジュール群１６はメモリのエリア獲得等の共通機能を有する共通機能プログラム１～ｘから構成され、保守運用プログラムモジュール群１８は保守運用機能を有する保守運用プログラム１～ｎから構成され、交換制御プログラムモジュール群１４は呼接続等の交換機の基本機能を有する交換制御プログラム１～ｍから構成され、サービス制御プログラムモジュール群１２は３者通話等のサービス提供機能を有するサービス制御プログラム１～ｋから構成される。また、サービス制御プログラムと交換制御プログラムは互いにリンケージを取りながら処理を実行する。
【０００４】従来の技術によると、オペレーティングシステムや保守運用機能を含めた、全ての交換サービスを制御する機能モジュールを上述した一つの交換ソフトウェア実行モジュールとして交換機にローデイングして運用している。従って、運用中の交換サービスに新しい機能を付加する場合、特に付加の規模が大きい場合や全く新規のサービス制御プログラムを追加しなけらばならない場合には、交換機システム全体のソフトウェアを稼働中の交換機システムを一時停止させて入れ替える方法がとられていた。また、付加する機能の規模が大きくない場合には、図２に示すように稼動中の交換機３０におけるサービス制御部３４あるいは交換制御部３２のプログラムの一部にパッチと呼ばれる部分変更を加えることで新しい機能を実現してきた。なお、サービス制御部３４では前述のサービス制御プログラムが実行され、交換制御部３２では前述の交換制御プログラムが実行される。図２中、パッチ生成環境４６にてソフトウェアパッチ４４が生成され、前述の保守運用プログラムの一部が実行されるパッチ管理部４２を介してパッチ領域３６に追加

サービスプログラム３８、４０が書き込まれ、交換制御部３２及びサービス制御部３４には、追加サービスプログラムに実行を移し、実行が終われば戻る旨のパッチ３３、３５が書き込まれる。

【０００５】

【発明が解決しようとする課題】しかしながら、上記の従来の方法によると、サービス追加の際に提供中のサービスや通話に影響を与えてしまうという問題点があった。すなわち、パッチによる機能追加の場合は、稼働中の交換機にパッチが投入された直後から追加されたプログラムが動作するため、呼処理データが変更される場合等には、パッチを投入する前に呼び出し中や通話中になっている呼がパッチ投入後に切断される可能性があった。また、パッチによると呼処理のデータを大幅に変更追加することが困難であるために、追加できる機能に制限がある。更に、パッチでは対応できない機能追加の場合には、新規プログラムを交換機全体のプログラムと共にコンパイルし直し、交換機を停止させて交換機のソフトウェア全体の入れ替えをする必要があった。

【０００６】本発明は上記の点に鑑みなされたもので、稼働中の交換機を停止させることなく、更にサービス実施中における呼び出し中・通話中等の呼状態に影響を与えずに、交換サービスを追加・変更・削除する交換機サービス制御方法及び交換機及び交換機サービス制御プログラムを記録した記録媒体を提供することを目的とする。

【０００７】

【課題を解決するための手段】上記課題を解決するための本発明の構成は、以下の通りである。請求項1 に記載の発明は、プログラムにより制御される交換機において、特定のサービスを実行するサービスプログラムがローディングされると、所定の場合に該サービスプログラムは該サービスプログラムが有する該サービスを実行するための関数へのポインタを第１のテーブルに記録し、該交換機は、発呼により該サービスの利用要求があると呼毎に一時的な記憶領域を確保し、該第１のテーブルを参照して得た該ポインタを該記憶領域内の第２のテーブルに保持し、該第２のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供することを特徴とする交換機サービス制御方法である。

【０００８】本発明により、動的に呼出関数へのリンケージをとることが可能となるので、、コンパイル時での呼び出し先と呼び出し元とのアドレスリンケージが不要となり、サービス追加等の場合に従来のようにパッチあるいは全体ファイル作成・入れ替えが不要となるため、サービス追加が容易となりサービスの中断も防止できる。また、サービス運用中にサービスプログラムを改造したサービスプログラムを再びローディングしても、サービス実施中の全く独立したサービスプログラムの呼はもとより、改造前のサービスプログラムの呼にも全く影響を与えずにサービスプログラムをローディングできる。

【０００９】請求項２に記載の交換機サービス制御方法は、前記交換機が、前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを有し、該データを用いて前記ローディングを制御する。本発明により、ローディングされるサービスプログラムの記憶領域が既存のサービスプログラムと衝突することを防止することができる。また、サービスプログラムを削除する場合にサービス実施中のサービスプログラムを不用意に削除できないよう制御することが可能になる。

【００１０】請求項３に記載の交換機サービス制御方法は、前記所定の場合が、前記交換機が周期的に前記ローディングを監視して前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である。本発明により、ローディングしたプログラムを迅速に使用することができる。

【００１１】請求項４に記載の交換機サービス制御方法は、前記サービスプログラムが独立してコンパイルされて前記ローディングされることである。本発明により、サービスプログラム開発が容易になる。請求項５に記載の交換機サービス制御方法は、既存のサービスプログラムを別のサービスプログラムに置き換える際に該既存のサービスプログラムを実行中の第１の呼が有る場合、該別のサービスプログラムによるサービスを提供すべき第２の呼が発生すると、前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持し、該ポインタの示す関数を実行することにより該第２の呼に該別のサービスプログラムによるサービスを提供する。

【００１２】本発明により、改造サービスの提供等の場合において、従来のように改造前のサービスを使用中の呼が切断されることなく、サービスを入れ替えることができる。請求項６に記載の交換機サービス制御方法は、ローディングされた前記サービスプログラムを前記交換機から削除する際、前記第１のテーブルの値を該サービスプログラムがローディングされる前の値に戻す請求項１記載の交換機サービス制御方法。

【００１３】本発明により、ローディングしたサービスプログラムを削除しても、呼の切断を発生させずにローディング前の状態でサービスを継続することが可能となる。以下の発明も上記と同様の効果を有する。請求項７に記載の発明は、プログラムにより制御される交換機であって、特定のサービスを実行するサービスプログラムをローディングする手段と、該サービスプログラムが有する該サービスを実行するための関数へのポインタを保持する第１のテーブルを有し、所定の場合に該サービスプログラムに該関数へのポインタを該第１のテーブルに

記録させる手段と、発呼により該サービスの利用要求がある場合に呼毎に一時的な記憶領域を確保する手段と、該第１のテーブルを参照して得た該ポインタを該記憶領域内の第２のテーブルに保持する手段と、該第２のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供する手段とを有することを特徴とする交換機である。

【００１４】請求項８に記載の交換機は、前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを有し、該データを用いて前記ローディングを制御する手段を有する。請求項９に記載の交換機は、前記交換機が周期的に前記ローディングを監視する手段とコマンド入力手段を有し、前記所定の場合が、該監視する手段により前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である。

【００１５】請求項１０に記載の交換機は、既存のサービスプログラムを別のサービスプログラムに置き換える際に既存のサービスプログラムを実行中の第１の呼が有る場合、該別のサービスプログラムによるサービスを提供すべき第２の呼が発生した場合に前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持する手段と、該ポインタの示す関数を実行することにより該第２の呼に該別のサービスプログラムによるサービスを提供する手段とを有する。

【００１６】請求項１１に記載の交換機は、ローディングされた前記サービスプログラムを前記交換機から削除する場合に前記第１のテーブルの値を該サービスプログラムがローディングされる前の値に戻す手段を有する。請求項1 2に記載の発明は、プログラムにより制御される交換機のサービスを制御するコンピュータ読み取り可能な交換機サービス制御プログラムを記録した記録媒体であって、該交換機に特定のサービスを実行するサービスプログラムをローディングし、所定の場合に該サービスプログラムに該サービスプログラムが有する該サービスを実行するための関数へのポインタを第１のテーブルに記録させ、発呼により該サービスの利用要求があると呼毎に一時的な記憶領域を確保し、該第１のテーブルを参照して得た該ポインタを該記憶領域内の第２のテーブルに保持し、該第２のテーブルに保持された該ポインタを参照することにより該ポインタの示す関数を実行してサービスを提供することを特徴とする交換機サービス制御プログラムを記録した記録媒体である。

【００１７】請求項１３に記載の交換機サービス制御プログラムを記録した記録媒体は、前記サービスプログラムがローディングされる記憶領域の領域空塞状態及び前記サービスプログラムの状態を管理するデータを保持し、該データを用いて前記ローディングを制御する。請求項１４に記載の交換機サービス制御プログラムを記録した記録媒体は、前記所定の場合が、前記交換機が周期的に前記ローディングを監視して前記サービスプログラムがローディングされたことを検出した場合、または、コマンドにより指示が入力された場合である。

【００１８】請求項１５に記載の交換機サービス制御プログラムを記録した記録媒体は、既存のサービスプログラムを別のサービスプログラムに置き換える際に既存のサービスプログラムを実行中の第１の呼が有る場合、該別のサービスプログラムによるサービスを提供すべき第２の呼が発生すると、前記交換機は新たに前記一時的な記憶領域を確保して該別のサービスプログラムの有する関数へのポインタのテーブルを該記憶領域に保持し、該ポインタの示す関数を実行することにより該第２の呼に該別のサービスプログラムによるサービスを提供する。

【００１９】請求項１６に記載の交換機サービス制御プログラムを記録した記録媒体は、ローディングされた前記サービスプログラムを前記交換機から削除する際、前記第１のテーブルの値を該サービスプログラムがローディングされる前の値に戻す。請求項1 7に記載の発明は、プログラムにより制御される交換機のサービスを制御するコンピュータ読み取り可能な交換機サービス制御プログラムを記録した記録媒体であって、該交換機に特定のサービスを実行させる関数と、該交換機にローディングされた場合に該関数へのポインタを該交換機内のテーブルに記録するプログラムを含み、該交換機により該テーブルに記録された該ポインタが参照されて該関数が実行されることを特徴とする交換機サービス制御プログラムを記録した記録媒体である。

【００２０】

【発明の実施の形態】図３は本発明の実施例である交換機５０における機能構成の概要を示す機能ブロック図である。同図に示すとおり、交換機５０は、交換制御を行う交換制御部５２、サービス制御部を起動するサービス起動部５４、サービスを実行するサービスプログラムであるサービス制御部５６及び７６、サービス制御部５８をシステム内に組み込むためのプラグインプラグアウト制御部６２、サービス制御部がローディングされる領域であるプラグインメモリ領域６１、プラグインメモリ領域６１におけるメモリの空塞状態を管理するプラグイン領域管理データ７８、サービス制御部を活性化させるためのサービス活性化制御部６６および活性化関数テーブル７０を有する。また、既存のサービス制御部を、改良したサービス制御部に置き換える場合や、サービス制御部を削除すなわちプラグアウトする場合にサービス実施中の呼に影響を与えないための機能であるリンケージ関数テーブル６０、呼処理インスタンス管理部７２、呼び出し関数テーブル７４を有する。また、新たなサービス制御部５８を生成するために、ファイル化環境６４が用いられる。

【００２１】次に、図４に示すフローチャートを用いて本実施例の動作の概要を説明する。ステップ１としてファイル化環境６４にてサービスの入れ替え、追加のためのサービス制御部５８を生成する。なお、サービス制御部は実行できる形式のプログラムであり、光ディスクや磁気テープ等の媒体に記録されている。ステップ２として、サービス制御部５８はプラグインプラグアウト制御部６２に読み込まれ、ステップ３として読み込まれたサービス制御部はプラグインメモリ領域６１の空きの領域に取り込まれる。ここで、プラグインメモリ領域６１のメモリの空塞状態はプラグイン領域管理データ７８により管理されており、プラグイン領域管理データ７８をプラグインプラグアウト制御部６２が参照することにより、サービス制御部がプラグインメモリ領域６１に取り込まれる。

【００２２】サービス制御部がローディングされると同時にステップ４として活性化関数テーブル７０に活性化用プログラムのエントリーアドレスが記録される。すなわち、サービス制御プログラムに組み込まれ、サービス制御部を活性化させるための関数（処理プログラム）のエントリーアドレスを活性化関数テーブル７０に記録する。なお、活性化とはサービス制御部がサービス起動部５４により起動がかけられる状態にすることをいう。

【００２３】ステップ５として、サービス活性化制御部６６が活性化関数テーブル７０に記録されたエントリーアドレスを参照し、サービス制御部７６に起動をかける。ステップ６として起動をかけられたサービス制御部７６は、サービスを実行する関数のポインタをリンケージ関数テーブル６０に記録する。ステップ７としてサービス起動部５４がリンケージ関数テーブル６０を参照する。ステップ８としてサービス起動部５４がリンケージ関数テーブル６０を参照して得られた関数のポインタを呼処理インスタンス管理部７２の呼び出し関数テーブル７４に記録する。そして、ステップ９としてサービス起動部５４は呼び出し関数テーブル７４に記録された関数ポインタを参照してサービス制御部における関数を実行してサービスを実行する。なお、呼び出し関数テーブル７４はダイヤルによりサービスがスタートした時点に呼処理インスタンス管理部７２に生成されるメモリ領域に設けられる。

【００２４】また、呼び出し関数テーブル７４はサービスの切り替え、追加時等に使用される。すなわち、改良サービスに切り替えるような場合であって、改良前のサービス制御部が発呼を受けて動作中であるとき、サービス起動部はリンケージ関数テーブルから改良サービスの関数のアドレスもらい、それを別の呼び出し関数テーブルに記録し、その後にそのサービスの発呼を受けた場合は、サービス起動部は新たな呼び出し関数テーブルを参照して改良サービスを実行する。サービスの切り替え、追加時等の動作の詳細については後述する。

【００２５】図５は本発明の実施例における交換ソフトウェア実行モジュールを示すモジュール図である。同図に示すように、交換ソフトウェア実行モジュール８０はプログラム基盤となるオペレーティングシステム８２、共通機能プログラムモジュール群８４、保守運用プログラムモジュール群８６、交換制御プログラムモジュール群８８、サービス起動プログラムモジュール群９０、リンケージ関数テーブル９２、実行モジュール９４_0～９４_nから構成される。なお、実行モジュール９４_0はオペレーティングシステム８２、共通機能プログラムモジュール群８４、サービス起動プログラムモジュール群９０、リンケージ関数テーブル９２をまとめた部分である。

【００２６】交換制御プログラムモジュール群８８、オペレーティングシステム８２、共通機能プログラムモジュール群８４、保守運用プログラムモジュール群８６、交換制御プログラムモジュール群８８は従来の技術と同様である。サービス起動プログラムモジュール群９０は、サービス起動プログラム１～ｎにより構成される。サービス起動プログラムは交換制御プログラムから起動がかかった後に、クロスリンケージ関数テーブル９２を参照しながら、サービス制御プログラムを決定し、起動する。実行モジュール９４_1～９４_nはサービス制御プログラムと関数テーブル生成プログラムから構成され、サービス制御プログラムは、実行するプログラムへの入り口の関数を関数表に備えている。また、活性化した場合に、関数テーブル生成プログラムは、リンケージ関数テーブルに入り口の関数のポインタを設定する。このようなソフトウェアモジュール構成をとることで、交換制御プログラム群を含む実行モジュール９４_0とサービス制御部の実行モジュール９４_1～９４_nを独立させ、サービスを新規に作成する場合や既存サービスを改造する場合など、既存部分である実行モジュール９４_0に影響を与えずに独立してコンパイルし実行モジュールを生成できる。従って、ファイル化環境６４において他のサービスとは独立したサービス制御部を得ることができる。

【００２７】図６は本発明の実施例である交換機のハードウェア構成図である。同図に示すように交換機１００は全体の制御を行う主制御装置（ＭＰＲ：Ｍａｉｎ　Ｐｒｏｃｅｓｓｏｒ）１１０、呼の制御を行う呼制御装置（ＣＰＲ：Ｃａｌｌ　Ｐｒｏｃｅｓｓｏｒ）１２０、１３０、１４０から構成される。主制御装置１１０は、プログラムの処理等を行う中央処理装置（ＣＰＵ）１１１、中央処理装置（ＣＰＵ）１１１に実行させるためのプログラム及びデータを記憶する主記憶装置１１２、交換機外部とのインターフェースをとるための入出力装置１１３、プログラム及びデータを記憶する外部記憶媒体１１４、光磁気ディスク等の可搬記憶媒体と情報をやりとりするための外部記憶装置１１５、各制御装置間で情報をアクセスする際の制御を行うプロセッサ間アクセス

制御装置１１６から構成される。呼制御装置１２０は外部記憶媒体１２１、プロセッサ間アクセス制御装置１２２、中央処理装置１２３、主記憶媒体１２４、通話路装置１２５から構成される。他の呼制御装置も同様の構成である。上記で説明した、本発明の実施例における機能は主に主制御装置１１０内の中央処理装置１１１において実行される。

【００２８】図７は、本発明の実施例である交換機５０の機能構成の詳細を示す機能ブロック図である。本実施例の主要部は前述した概要構成と同様であるので各部の詳細を説明する。サービス起動部５４はｎ個のサービス起動プログラム５５$_1$～５５$_n$を内蔵している。呼処理インスタンス管理部７２は、起呼時に呼処理で必要なメモリブロックを捕捉する。このメモリブロック内には、呼毎に呼び出し関数テーブル７４が割りつけられる。サービス制御部５６にはサービス制御プログラム５７と関数テーブル生成プログラム６７が内蔵されており、サービス制御部７６にはサービス制御プログラム７７と関数テーブル生成プログラム７９が内蔵されている。プラグインプラグアウト制御部６２はプラグインメモリ領域管理データ７８、サービス制御部をローディングするPlug-in Loader６９及びサービス制御部がローディングされるプラグインメモリ領域（図７には記載を略している）内の領域をプラグイン領域管理データを参照して決定するPlug-in Manager ６８を有している。サービス活性化制御部６６は活性化プログラム６５を有し、サービス活性化要求があった場合に活性化関数テーブル７０を参照する。

【００２９】以下、図７及びフローチャート等を用いて本発明の実施例の動作について詳細に説明する。図８は、図７に示す実施例において、サービスプラグインと称しているサービス制御部のローディングについての処理を示すフローチャートである。ファイル化環境６４で生成されたサービス制御部５８は、ステップ１として外部記憶装置１１５から読み出され（図７（ａ））、ステップ２として一旦外部記憶媒体１１４に保存される。ステップ３としてPlug－in Loader ６９はPlug-in Manager ６８にローディング先のアドレスを問い合わせる。ステップ４としてPlug-in Manager ６８は、サービス制御部がリローケータブル実行形式すなわちどのメモリアドレスでも動作可能な実行形式であるかどうかを調べ、リロケータブルであればステップ５としてプラグイン領域管理データ７８を基にして空き領域を補足する。ステップ４においてサービス制御部がリローケータブル実行形式でない場合、例えば絶対番地実行形式すなわちコンパイル時点で実行するメモリアドレスが決定されている形式の場合には、ステップ６としてプラグイン領域管理データ７８を基にしてPlug-in Manager ６８によってコンパイラが割りつけた実行アドレスがプラグインメモリ領域の空エリアであるかどうかが検証される。ステップ７として、検証結果が二重割り付けかどうかが判断され、二重割り付けであればステップ８として警告メッセージが出力されプラグインが拒否される。ステップ７において二重割り付けでなければステップ９に進む。ステップ５とステップ８の後もステップ９に進む。ステップ９においてサービス制御部のメモリ割付が完了したかどうかがチェックされ、完了していない場合は、サービスプラグイン処理を終了する。完了している場合は、ステップ１０として、サービス制御部をロードするメモリ領域アドレスを決定し、ステップ１１として、割り付け状態を表す割り付け状態テーブルを”プラグイン済”の状態で設定する。

【００３０】ステップ３においてアドレス問い合わせが終了すると、ステップ１２として、メモリ領域が二重割り付けかどうかチェックし、二重割り付けならば処理を終了する。二重割り付けでなければ、ステップ１３として、捕捉されたメモリ領域アドレスにサービス制御部をロードし（図７（ｂ））、ステップ１４としてロードしたサービス制御部の関数テーブル生成プログラムのアドレスが活性化関数テーブル７０に書き込まれ（図７（ｃ））処理を終了する。

【００３１】図９はプラグイン領域管理データ７８の例であり、プラグインメモリ領域の空塞管理はビットマップ形式の（ａ）に示すようなメモリ空塞テーブルで管理される。このテーブルでの1 ビットは8KB のエリアを意味し、”１”が使用中、”0”が空きを示す。サービス制御部がリロケータブル実行形式の場合、プラグインするサービス制御部のサイズを8KB で割り、格納可能なエリアサイズを求め、連続して空いているエリアすなわち連続して”０”が存在するビット位置を求めることにより格納アドレスを決定する。サービス制御部が絶対番地実行形式である場合には、コンパイラが割り付けたサービス制御部の先頭番地からメモリ空塞テーブルのビット位置をもとめ、そのサイズ分が空きエリアであるかをチェックする。

【００３２】図９（ｂ）はプラグイン領域管理データの割り付け状態テーブルを示し、システムに実装可能な最大サービス分の情報が管理できるようにあらかじめエリアが確保されている。このテーブルは主にプラグインメモリ領域の状態表示機能に使用される。例えば、プラグインメモリ領域にどのようなサービスが実装されているかやプラグインメモリ領域の使用状況を表示するコマンド等で利用されている。このため、割り付け状態テーブルにはサービス名称やプラグインメモリ領域位置やその専有サイズ、さらにそのサービスの状態を保持している。また、この割り付け状態テーブルに前述のリロケータブル形式か絶対番地形式かのプログラム属性の情報を持つことで、プラグインメモリ領域の空きエリアが点在してしまった場合等、例えばプラグイン／プラグアウトを繰り返すことによりメモリ空塞テーブル上空きビット

が点在してしまった場合等に、リロケータブル形式のサービス制御部を再割り付けし、空きエリアを拡大することができる。

【００３３】以上の処理によって、サービス制御部５８はプラグインプラグアウト制御部６２でプラグインメモリ領域上にローデイングされる。図１０に活性化関数テーブル７０の例を示す。活性化関数テーブル７０には、サービス制御部の識別番号であるサービス番号によって得られるサービス制御部内の関数テーブル生成プログラムへのアドレスが設定される．このテーブルは後述するサービス活性化制御部６６で参照される。

【００３４】図１１は、サービス制御部の活性化とリンケージ関数テーブル生成の処理を示すフローチャートである。ステップ１としてサービス制御部がプラグインメモリ領域にローディングすなわちプラグインされると、ステップ２としてサービス活性化制御部６６は活性化関数テーブル７０より、ロードされたサービス制御部内の関数テーブル生成プログラムのアドレスを得て、ステップ３として関数テーブル生成プログラムを起動（図７（ｄ））して関数テーブル生成処理を開始し、ステップ４としてリンケージ関数テーブルを読み出す。ステップ５としてリンケージ関数テーブル６０に設定されたサービスが新規サービスかどうかをチェックし、新規サービスならばステップ７に進み、新規サービスでなければ、ステップ６として旧リンケージ関数テーブルの内容を保存してからステップ７に進む。ステップ７において、サービス制御プログラムの全関数をリンケージ関数テーブルに設定する（図７（ｅ））。ステップ８として設定完了かどうかをチェックし、完了していなければステップ９として設定異常警告を発して処理を終了し、設定が完了していればそのまま処理を終了する。

【００３５】ステップ３により関数テーブル生成プログラムを起動すると、ステップ１０として関数テーブルの生成が完了したかどうかをチェックし、完了していなければ処理を終了し、完了していればステップ１１としてプラグイン領域管理データの割り付け状態テーブルにおける割付状態を”実装サービス中”に変更して処理を終了する。

【００３６】サービス活性化、リンケージ関数テーブル生成について図７を用いて更に説明する。図７のサービス制御部５６、７６は、他の実行モジュールと独立してコンパイルされているため、プラグインメモリ領域上にローデイングされても他の実行モジュールからは呼び出すことができない。このため、サービス活性化制御部６６の活性化プログラム６５によりサービス制御部７６を活性化すなわち使用可能状態にするのである。

【００３７】活性化プログラム６５は周期起動またはコマンドにより起動される。周期起動の場合には、定期的にプラグインプラグアウト制御部６２内のプラグインメモリ領域管理データ７８を監視することで、プラグインされたサービス制御部７６のサービス番号を知ることができる。また、コマンドで起動する場合には直接保守者からサービス番号を入力するか、または入力されたサービス名称からプラグインメモリ領域管理データ７８を利用してサービス番号を得ることができる。このようにして得たサービス制御部７６のサービス番号と活性化関数テーブル７０から、サービス制御部７６内の関数テーブル生成プログラム７９に起動をかける（図７の（ｄ））。活性化プログラム６５から起動された関数テーブル生成プログラム７９は、サービス制御プログラム７７内に定義されている関数表をもとにリンケージ関数テーブル６０に関数のアドレスを設定するのである（図７の（ｅ））。

【００３８】図１２は、リンケージ関数テーブル６０の構成と関数テーブル生成プログラム７９の動作を示した図である。図１２において、リンケージ関数テーブル６０はサービス番号でインデックスされる初段テーブルと関数番号でインデックスされる関数ポインタテーブルで構成されている。初段テーブルには、関数ポインタテーブルのアドレスが書かれており、関数ポインタテーブルには、サービス制御プログラム７７の関数ポインタアドレスが設定できるようになっている。サービス制御部７６内の関数テーブル生成プログラム７９は、コンパイル時に生成されているサービス制御プログラム７７の関数表およびサービス番号をもとにリンケージ関数テーブル６０の関数ポインタテーブルに関数ポインタを設定する。このように設定されたリンケージ関数テーブル６０はサービス起動プログラムによって参照される。

【００３９】図７において、あるサービスが起動された場合に交換制御部５２はそのサービスに対応する特定のサービス起動プログラム、ここではサービス起動プログラム55_2を呼び出す（図７の（ｆ））。サービス起動プログラム55_2はまず呼処理インスタンス管理部７２からメモリブロックを捕捉する。その後、前述のように設定されたリンケージ関数テーブル６０から関数情報を読み出し（図７の（ｈ））、捕捉されたそのメモリブロック内の呼出関数テーブル７４に関数アドレスを保存する（図７の（ｇ））。サービス起動プログラム55_2からサービス制御プログラム７７を起動する場合（図７の（ｉ））には、この呼出関数テーブル７４が使用される。

【００４０】すなわち、交換制御部５２から起動されるサービス起動プログラムは、直接サービス制御プログラムの関数を呼び出すのではなく、サービス番号に従ってリンケージ関数テーブルから取得した関数ポインタテーブル１５０（図１２）を、一旦呼処理インスタンス管理部７２内の呼制御用メモリブロックに呼び出し関数テーブル７４として保存し、その保存した呼び出し関数テーブルを参照してサービス制御プログラムの関数１６０（関数１～３）を呼び出す。

15

【００４１】図１３は上記の処理をフローチャートで示した図である。同図において、あるサービスの起呼時、ステップ１として交換制御部からサービス起動プログラムに起動がかかり、ステップ２としてサービス起動プログラムは呼処理インスタンス管理部７２のメモリエリアを捕捉する。ステップ３としてリンケージ関数テーブルから関数のアドレスを読み出し、ステップ４としてその読み出したアドレスを、呼処理インスタンス管理部７２のメモリエリア内の呼制御メモリブロックにコピーすることにより呼び出し関数テーブル７４を設定する。ステップ５としてサービス起動プログラムは呼び出し関数テーブル７４を読み出す。また、サービスにより、ダイアル数字受信、接続応答、切断等に応じて、呼処理インスタンスメモリエリア内の読み出し関数テーブルを読み出す。ステップ６として、読み出した関数アドレスが有効かどうかをチェックし、有効でなければステップ７として呼の強制開放等の異常時処理を行い処理を終了し、有効であればステップ８としてサービス制御プログラムの起動を行い処理を終了する。ステップ８においては、ダイアル数字受信、接続応答、切断等の受信信号の種類に応じて呼び出し関数テーブルから決定されたサービス制御部のサービス制御プログラムが呼び出される。

【００４２】上述の通り本発明によれば、

（１）全く新しいサービスとしてのサービス制御部を追加する場合であっても、リンケージ関数テーブル７２に新しいサービス制御プログラムの関数を設定することで機能追加ができる。

（２）既存サービスを改造する場合であっても、改造したサービス制御部を新しいサービス同様に追加してリンケージ関数テーブルを書き換えることにより新旧を入れ換えることができる。この時、古いサービス制御部で動作している呼があっても発呼時にサービス起動プログラムが補足した呼制御用メモリブロック内にサービス制御部の関数ポインタが保存されているため、通話中の呼に影響を及ぼすことなく処理を実行できる。

【００４３】以上の一連の処理の流れを、フリーダイアルサービスがある加入者から発呼された場合を例にとり図１４～１６を用いて説明する。まずステップ１としてシステム初期設定を行う。すなわち図１４に示すように、サービス制御部内の関数テーブル生成プログラムがシステムの初期設定時にサービス制御プログラムに実装されたフリーダイアルサービスを実行する関数（ｆ１とｆ２）のポインタアドレスをリンケージ関数テーブルに設定する。

【００４４】図１５において、フリーダイアルサービスがある加入者から発呼された時、ステップ２として、交換制御プログラムからサービス起動プログラムに起動がかけられ、ステップ３としてサービス起動プログラムが呼処理インスタンス管理部内で呼制御メモリブロックを補足し、ステップ４としてリンケージ関数テーブルの内容を呼制御メモリブロックにコピーする。呼制御メモリブロックは、その呼が終了するまでの間その内容を保持できるメモリ領域である。ステップ５として呼制御メモリブロックに保持された関数テーブルを参照し、ステップ６としてサービス起動プログラムからサービス制御プログラムが実行される。

【００４５】ここで、更に機能追加したフリーダイアルサービスが追加される場合を図１６に示す。サービス制御プログラムとしてローディングされた新たなフリーダイアルサービスをフリーダイアル２と称し、関数としてはｆ１’とｆ２’を有することとする。フリーダイアル２に対応するサービス制御プログラムをローデイングした時に関数テーブル生成プログラムが起動され、ステップ７として、追加された関数ｆ１’とｆ２’のエントリアドレスがリンケージ関数テーブルに上書き設定される。呼処理インスタンス管理部内の呼制御メモリブロック上に前のサービス制御プログラムへの関数（f1とｆ２）が保持され、サービス制御プログラムがステップ８として参照しているため、既にサービスを開始している加入者は、新サービスプログラムがローディングされても影響を受けない。

【００４６】再び新しい加入者からの発呼を交換制御プログラムが受け付け、ステップ９としてサービス起動プログラムが起動された場合には、ステップ１０としてサービス起動プログラムによって新しい呼制御メモリブロックが補足され、ステップ１１としてｆ１’とｆ２’を含む関数テーブルが呼制御メモリブロックにコピーされ、ステップ１２としてサービス起動プログラムが関数へのアドレスを参照して、ステップ１２として呼処理サービスが実行される。この場合、新旧のサービス制御プログラムが同時に動作しているが、やがて旧サービスプログラムを使用している加入者が終話することで古いサービス制御プログラムが使用されなくなる。

【００４７】以上、サービスの追加、切り替え、すなわちサービスプラグインについて説明した。次に、一度プラグインしたサービス制御部をプラグイン前の状態に戻すサービスプラグアウトについて説明する。プラグインしたサービス制御部を単にプラグアウトするだけでは実行プログラムが削除されサービスが停止するばかりか、システムに重大な影響を与えることになる。そこで、プラグアウト時には、下記の手順に従う。

【００４８】（１）プラグアウトすべきサービス制御部への発呼を停止する。

（２）既に通話中すなわちサービス実施中の呼を監視し、すべての呼が終話した時点でサービス制御プログラムをプラグアウトする。

具体的なプラグアウト処理の手順を図１７のフローチャートおよび図７の機能ブロック図を用いて説明する。

【００４９】プラグアウト時には、ステップ１としてサービス活性化制御部６６内にサービスの閉塞をコマンド

で通知する。ステップ２として活性化プログラム６５は閉塞通知をPlug-in Manager ６８に渡す（図７（ｊ））。ステップ３としてPlug-inManager ６８はプラグイン領域管理データ７８内の割り付け状態テーブルの状態値を”プラグアウト要求中”に書き換える。ステップ４として、Plug-in Manager ６８は定期的にこの割り付け状態テーブルを検索し、プラグアウト要求中のサービスについて現在通話中の呼が存在するかを呼処理インスタンス管理部７２に問い合わせる（図７（ｋ））。ステップ５として通話中呼がないかを判定し、通話中呼があればステップ６としてしばらく待ってからステップ５に戻り、通話中呼が存在しない場合には、ステップ７としてプラグイン領域管理データ７８内の割り付け状態テーブルの該当サービスの状態値を初期値に戻し、ステップ８として閉塞が完了したかどうかをチェックし、完了していなければ処理を終了し、完了していればステップ９として、呼び出し関数テーブルのリカバリとしてステップ１０からの処理を行う。

【００５０】ステップ１０として活性化プログラム６５は関数テーブル生成プログラム７９を起動する。ステップ１１として関数テーブル生成プログラム７９は旧リンケージ関数テーブルの読み出しを行い、ステップ１２としてテーブルがあるかどうかをチェックし、テーブルがなければステップ１３としてリンケージ関数テーブルを初期設定し、テーブルがあればステップ１４として旧リンケージ関数テーブルの内容でリンケージ関数テーブルを設定して、関数テーブルのリカバリの処理を終了する。すなわち、関数テーブル生成プログラムは該当するリンケージ関数テーブル６０の関数ポインタを書き換え前の状態（新規サービスであれば〃０〃に、既存サービス入れ換えであれば既存サービスの関数ポインタ値）に設定するのである。したがって、交換制御部５２からサービス起動プログラムが起動された時に、関数ポインタ値が〃０〃であればサービス制御プログラムは起動されず、新しく発呼される呼は規制される。また、既存サービスへのリンケージ関数テーブルの値が関数ポインタである場合には、そのポインタを使用して既存サービスが起動されるため、サービスの中断が発生しない。

【００５１】次に、ステップ１５として活性化関数テーブル７０のリカバリを行い、ステップ１６としてサービス制御部をプラグインメモリ領域上から取り除く。交換機サービス制御プログラムを記録した記録媒体は図６に示す主記憶装置１１２又は外部記憶媒体１１４のいずれか一方または両方に相当するものである。上記の記録媒体には上述した説明の手順に従い中央処理装置１１１を動作させるためのプログラムが格納されており、既存の交換機のメモリ部等にこのプログラムを格納することにより、既存の交換機を本発明の交換機として使用することができる。主記憶装置１１２又は外部記憶媒体１１４を実現する記録媒体としては、電子メモリやハードディスク等がある。また、本発明の交換機サービス制御プログラムを格納した光磁気ディスク、磁気テープ等から既存の交換機のメモリ部等にこのプログラムをローディングすることにより既存の交換機を本発明の交換機として使用することができる。

【００５２】なお、本発明は、上記の実施例に限定されることなく、特許請求の範囲内で種々変更・応用が可能である。

【００５３】

【発明の効果】本発明により、交換機の動作中でのサービスの入れ替えが可能になる。従来はサービスを入れ替えるためにシステム全体のファイル生成及びシステムスタート時の再開（初期化）動作が必要となり、サービスの一時中断が発生していたが、本発明により、交換機全体の動作に影響を与えることなく、該当するサービスの切り替えおよび追加を可能になる。また、今後の交換機の技術動向の一つとなり得る「ユーザ・プログラマビリティ」の実現のために本発明は有効である。すなわち、これまで交換機ユーザ（電話会社等）は加入者付加サービスなどの新機能追加作業を交換機ベンダーに依存して行っていたが、本発明のようにサービスに影響を与えることなく機能追加できることとなれば、ユーザ自身が新機能の開発を行ってサービス機能追加をより短期間で行うことが可能となる。本発明はその基盤技術となり得るものである。

【図面の簡単な説明】

【図１】従来技術による交換ソフトウェア実行モジュールを示すモジュール図である。

【図２】従来技術においてパッチを用いてサービス機能を追加する例である。

【図３】本発明の実施例における交換機の機能構成概要を示す機能ブロック図である。

【図４】図３に示す構成の動作を示すフローチャートである。

【図５】本発明の実施例における交換ソフトウェア実行モジュールを示すモジュール図である。

【図６】本発明の実施例における交換機の構成を示すブロック図である。

【図７】本発明の実施例における交換機の機能構成詳細を示す機能ブロック図である。

【図８】図７に示す実施例において、サービス制御部のローディングについての処理を示すフローチャートである。

【図９】プラグイン領域管理データ７８の例である。

【図１０】活性化関数テーブル７０の例である。

【図１１】サービス制御部の活性化とリンケージ関数テーブル生成の処理を示すフローチャートである。

【図１２】リンケージ関数テーブル６０の構成と関数テーブル生成プログラム７９の動作を示した図である。

【図１３】あるサービスの起呼が発生した場合の交換制

御部とサービス起動プログラムによる処理を示したフローチャートである。

【図１４】初期設定を示す図である。

【図１５】フリーダイアルサービスがある加入者から発呼された場合を示す図である。

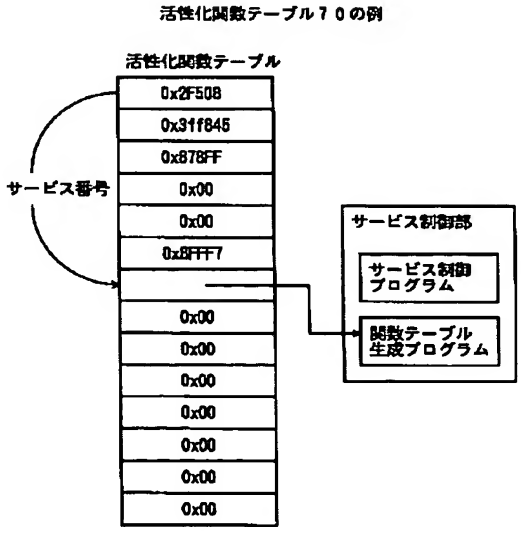
【図１６】新たに機能追加したフリーダイアルサービスが追加される場合を示す図である。

【図１７】プラグアウト処理の手順を示すフローチャートである。

【符号の説明】

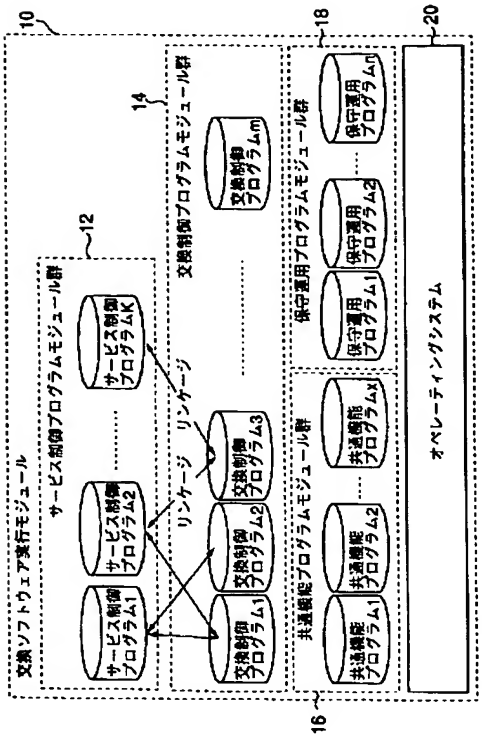
１０、８０　交換機ソフトウェア実行モジュール
１２　サービス制御プログラムモジュール群
１４、８８　交換制御プログラムモジュール群
１６、８４　共通機能プログラムモジュール群
１８、８６　保守運用プログラムモジュール群
２０、８２　オペレーティングシステム
３０　交換機
３２　交換制御部
３３、３５　パッチ
３４、５６、５８、７６　サービス制御部
３６　パッチ領域
３８、４０　追加サービスプログラム
４２　パッチ管理部
４４　ソフトウェアパッチ
４６　パッチ生成環境
５０、１００　交換機
５２　交換制御部
５４　サービス起動部
５５　サービス起動プログラム
５７、７７　サービス制御プログラム
６０、９２　リンケージ関数テーブル
６１　プラグインメモリ領域
６２　プラグインプラグアウト制御部
６４　ファイル化環境
６５　活性化プログラム
６６　サービス活性化制御部
６７　関数テーブル生成プログラム
６８　Plug-in Manager
６９　Plug-in Loader
７０　活性化関数テーブル
７２　呼処理インスタンス管理部
７４　呼び出し関数テーブル
７７　サービス制御プログラム
７８　プラグイン領域管理データ
７９　関数テーブル生成プログラム
９０　サービス起動プログラムモジュール群
９４　実行モジュール
１１０　主制御装置
１１１、１２３　中央処理装置
１１２、１２４　主記憶装置
１１３　入力装置
１１４、１２１　外部記憶媒体
１１５　外部記憶装置
１１６、１２２　プロセッサ間アクセス制御装置
１２０、１３０、１４０　呼制御装置
１２５　通話路装置
１５０、１６０　関数１～３

【図１０】



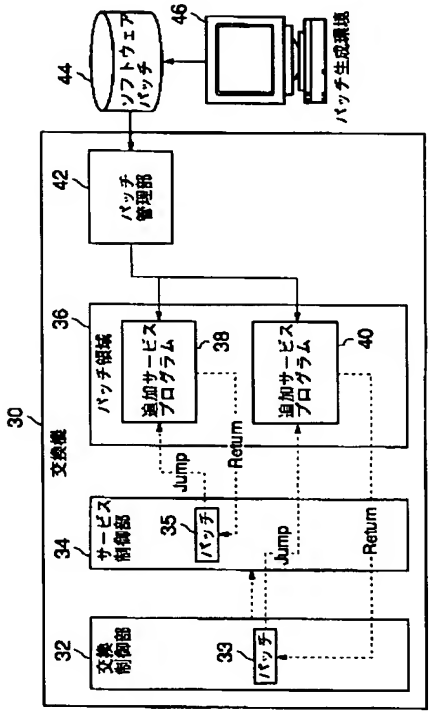
【図１】

従来技術による交換ソフトウェア実行モジュールを示すモジュール図



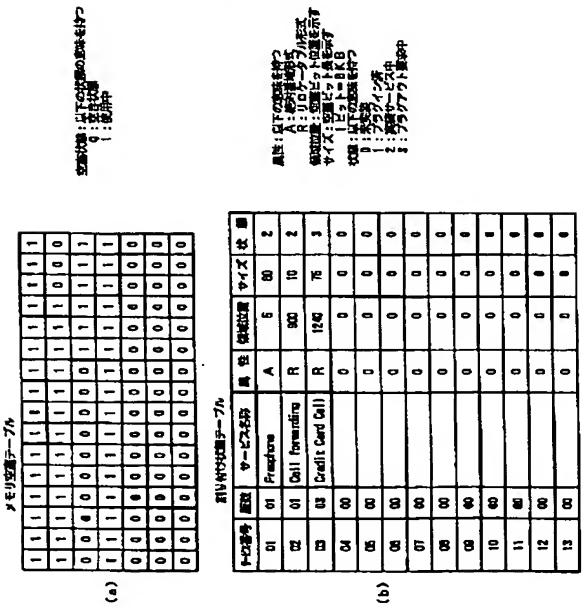
【図２】

従来技術においてパッチを用いてサービス機能を追加する例



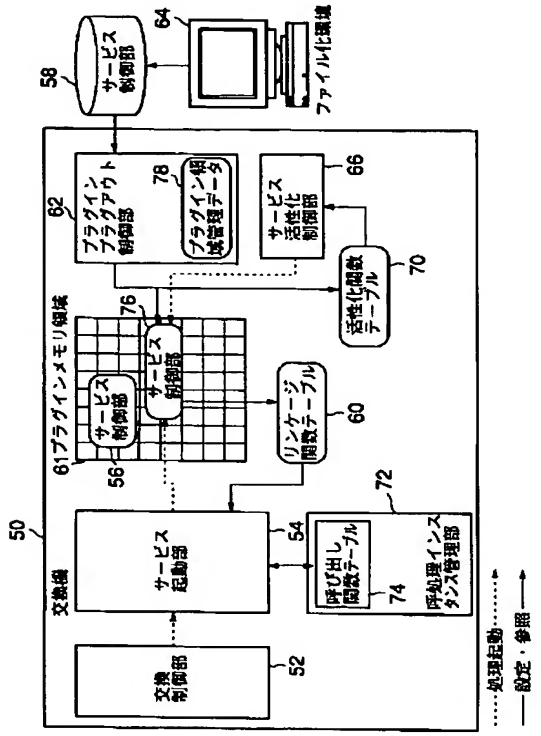
【図９】

プラグイン領域管理データ７８の例



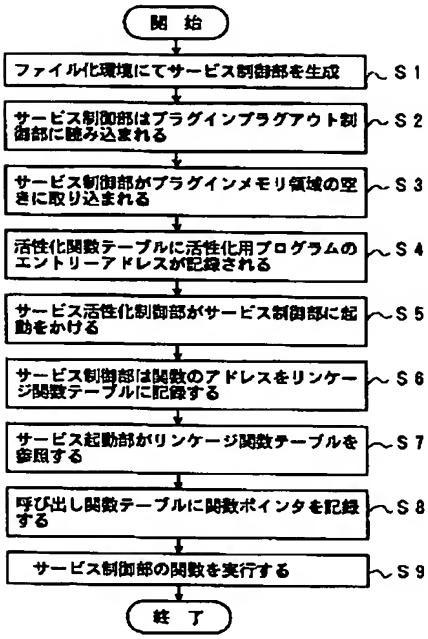
【図３】

本発明の実施例における交換機の機能構成概要を示す機能ブロック図



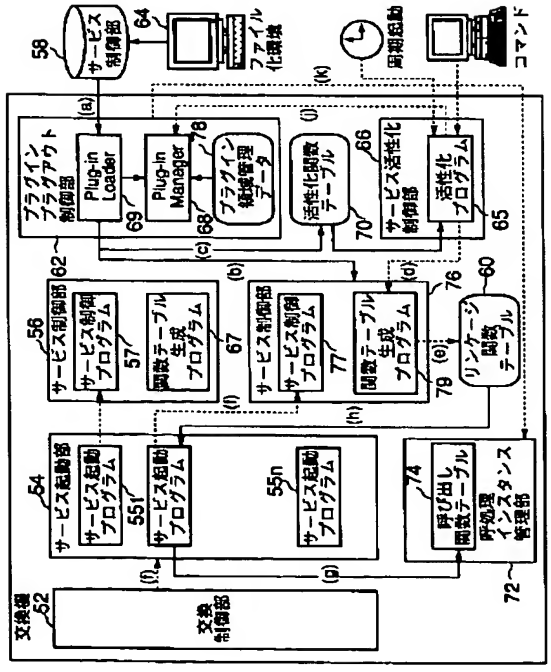
【図４】

図３に示す構成の動作を示すフローチャート



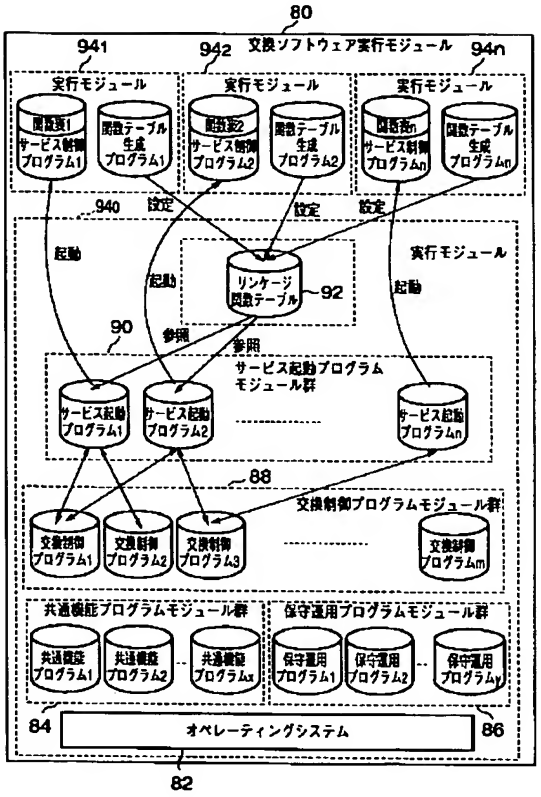
【図７】

本発明の実施例における交換機の機能構成詳細を示す機能ブロック図



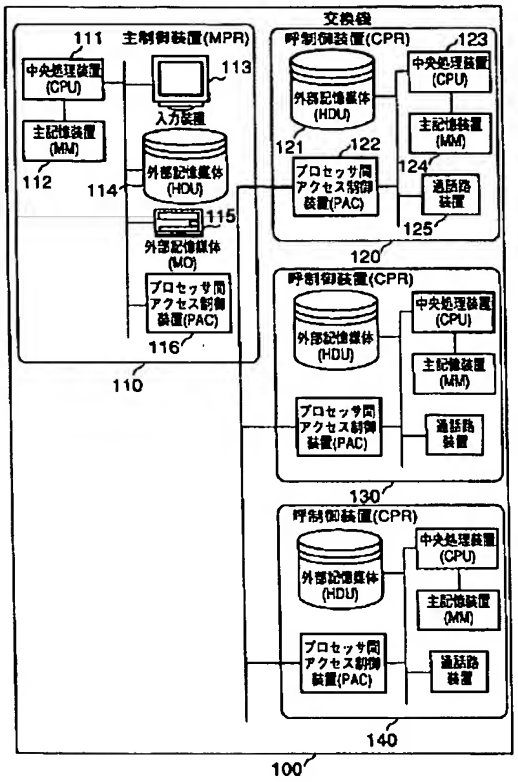
【図５】

本発明の実施例における交換ソフトウェア実行モジュールを示すモジュール図



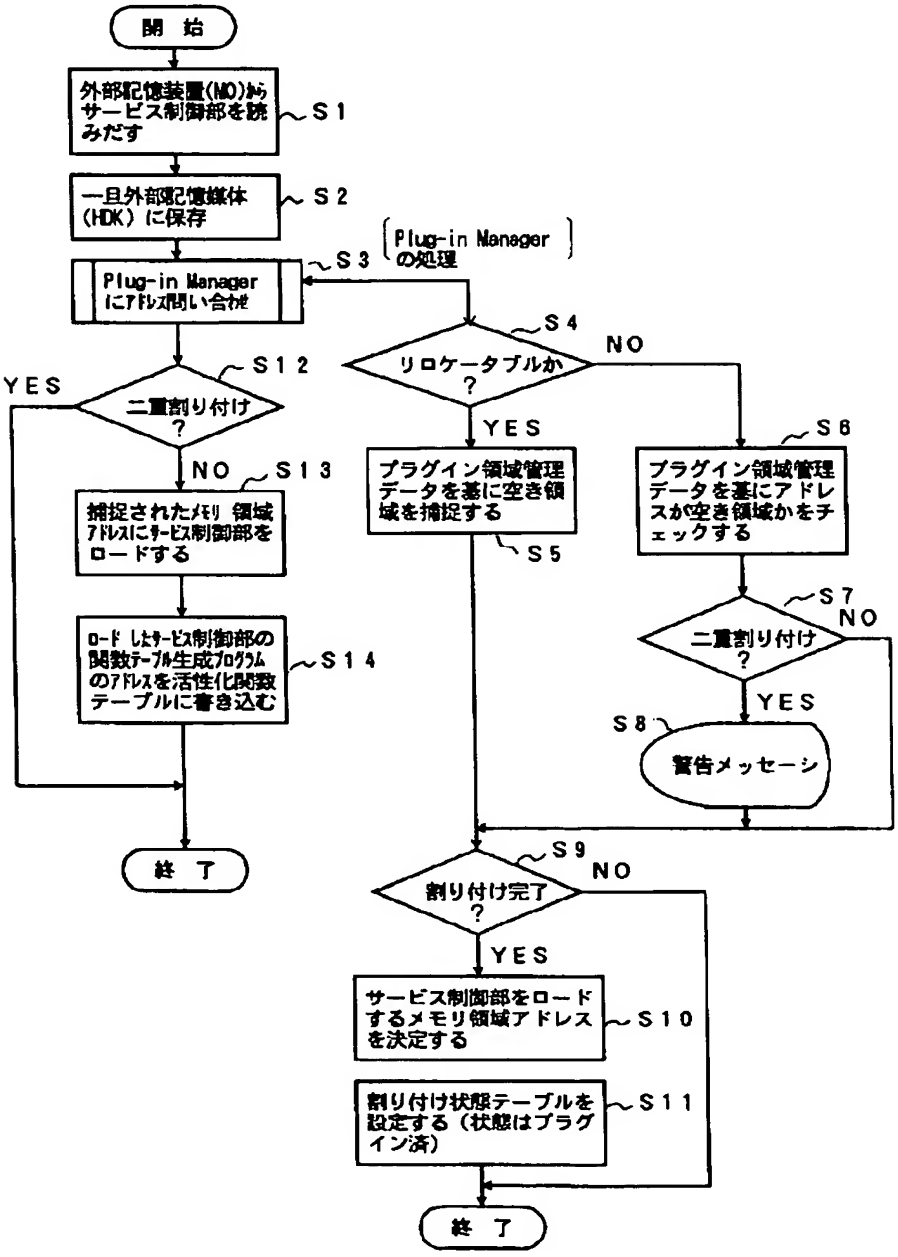
【図６】

本発明の実施例における交換機の構成を示すブロック図



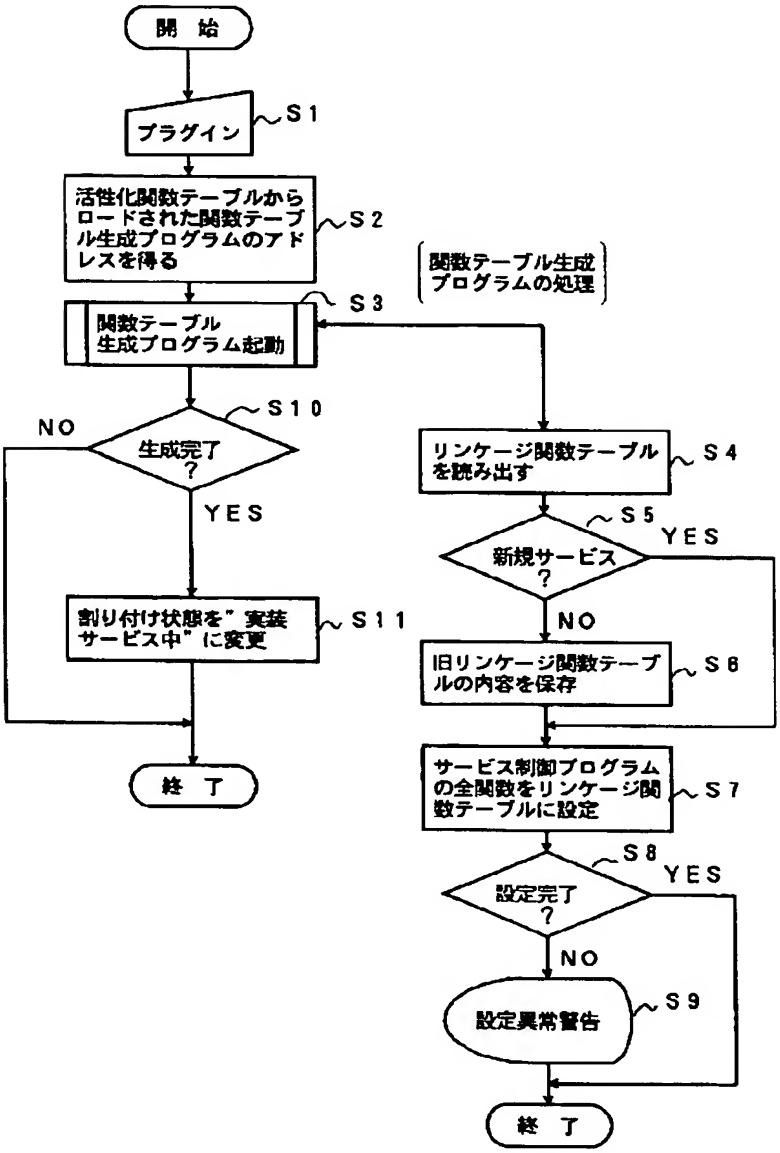
【図８】

図７に示す実施例において、サービス制御部のローディングについての処理を示すフローチャート



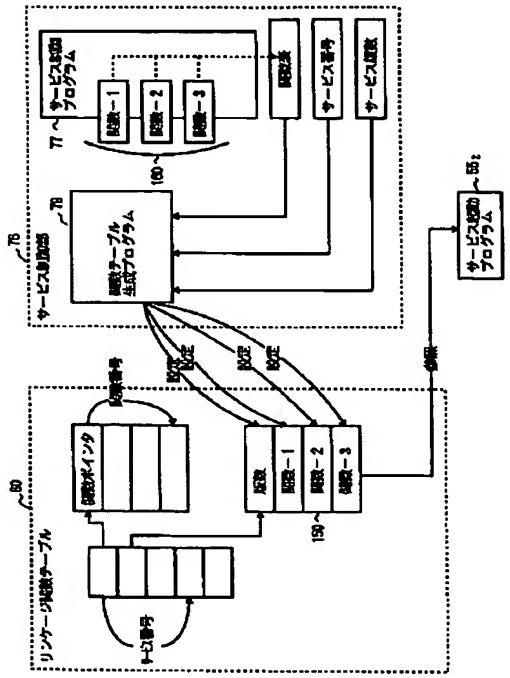
【図１１】

サービス制御部の活性化とリンケージ関数テーブル生成の処理を示すフローチャート



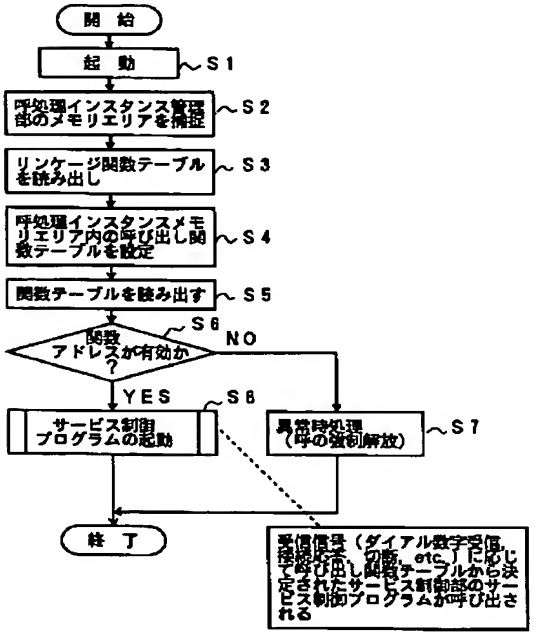
【図１２】

リンケージ関数テーブル６０の構成と関数テーブル生成プログラム７９の動作を示した図



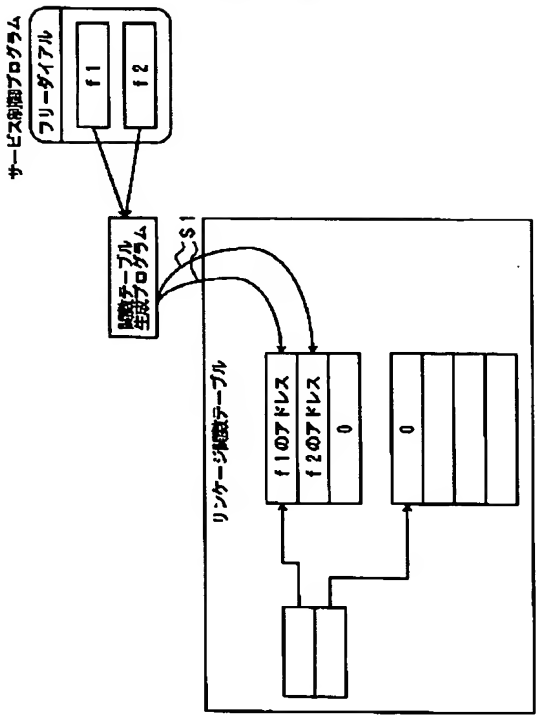
【図１３】

あるサービス起呼が発生した場合の交換制御部とサービス起動プログラムによる処理を示したフローチャート



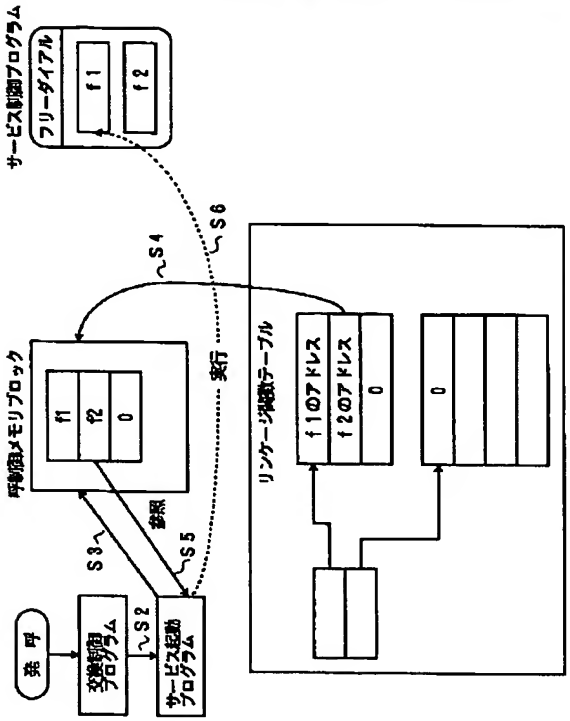
【図１４】

初期設定を示す図



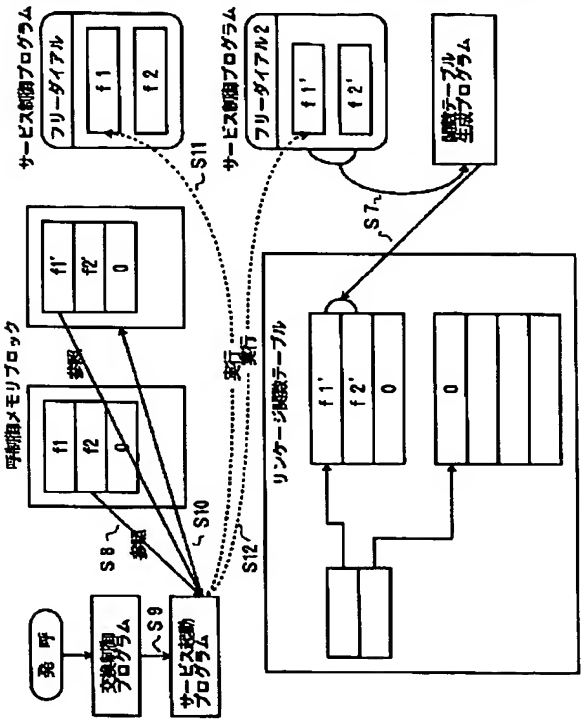
【図１５】

フリーダイアルサービスがある加入者から発呼された場合を示す図



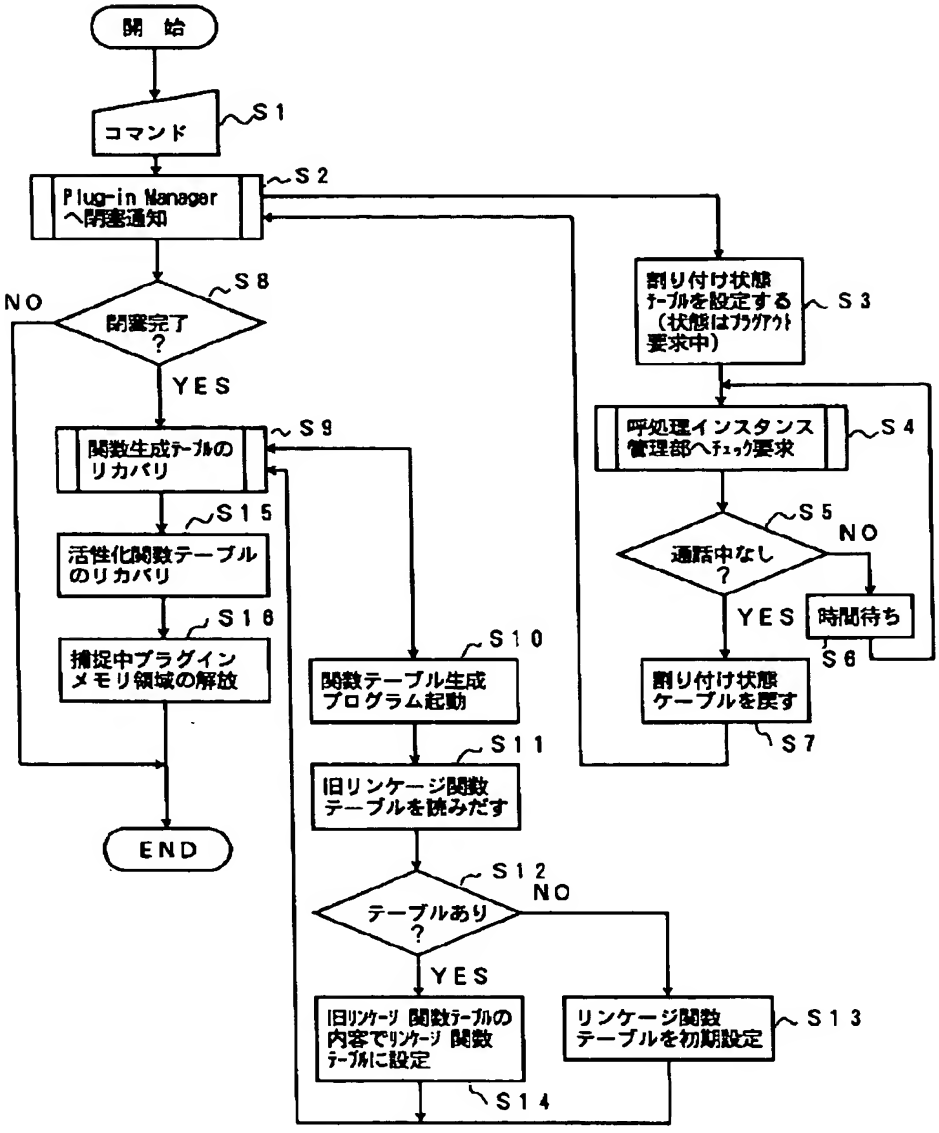
【図１６】

新たに機能追加したフリーダイアルサービスが追加される場合を示す図



【図１７】

プラグアウト処理の手順を示すフローチャート



フロントページの続き

(72)発明者　加室　真吾
神奈川県川崎市中原区上小田中４丁目１番
１号　富士通株式会社内

Ｆターム(参考)　5B076 EA17
5K024 AA00 AA45 BB04 GG00
5K026 AA10 BB00 CC07 FF02 FF03
FF04 FF23 GG12 GG15
9A001 CC02 CC07 JJ12 LL05 LL09